MC-Fluid: rate assignment strategies

Saravanan Ramanathan and Arvind Easwaran

Nanyang Technological University, Singapore

December 1, 2015

NANYANG TECHNOLOGICAL UNIVERSITY

Outline

Outline

Mixed-Criticality (MC) Task Model

Implicit Deadline Sporadic Task : $\tau_i = (T_i, L_i, C_i)$

- T_i is the minimum separation between successive job releases
  - Since we consider implicit deadline tasks, deadline = T_i
- L_i denotes the criticality level of task (assume 2 levels)
  - LO denoting low-criticality and HI denoting high-criticality
- $C_i = \{C_i^L, C_i^H\}$: C_i^L denotes LO worst-case execution time (WCET), and $C_i^H (\geq C_i^L)$ denotes HI WCET
  - $C_i^H = C_i^L$ if $L_i = LC$

Mixed-Criticality (MC) Scheduling

Task system behaviours : A MC task system with two criticality levels can exhibit the following behaviours

- LO mode: The system is in this behaviour as long as no task has executed beyond its LO WCET
- HI mode: The system switches to this behaviour when any HI task executes beyond its LO WCET

MC Correctness: A MC system is said to be correct if

- In LO mode: All tasks with LO WCETs are schedulable
- In HI mode: Only HI tasks with HI WCETs are schedulable
  - All LO tasks are dropped

Fluid Scheduling

Fluid Scheduling: Each task is assigned a fractional processing capacity at each time instant

- Schedulability: A task τ_i can meet its deadline if
  - Rate (θ_i) x Period $(T_i) \geq$ WCET
- Feasibility: A task rate θ_i is valid under a m core system if
  - $\theta_i \leq 1$
  - $\sum_{\tau_i \in \tau} \theta_i \leq m$

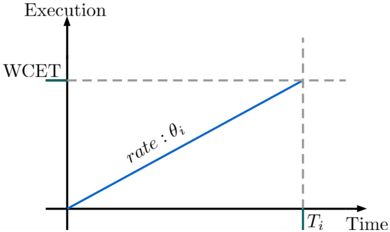


Figure: Fluid scheduling

MC-Fluid Scheduling

MC-Fluid Platform: Each task is executed with LO-rate (θ_i^L) in LO mode and HI-rate (θ_i^H) in HI mode

- At mode switch, execution requirement is changed
- Execution rate is changed
- Carry-over job: A job released in LO mode and finished in HI mode

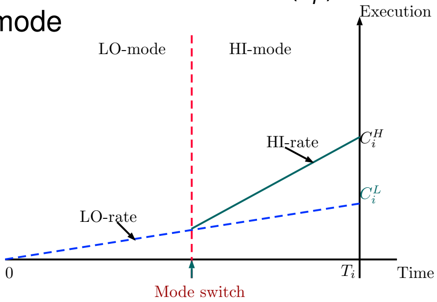


Figure: Carry-over job

MC-Fluid Scheduling

Rate Assignment:

- Worst-case mode switch pattern
  - Minimum $(\theta_i^L - u_i^L)$
- Construct an optimization problem
  - Solve it by convex optmization
- Optimal rate assignment algorithm
  - Schedulable rate assignment for all feasible task sets
- Has polynomial complexity

MC-Fluid Scheduling

- θ_i^H is determined by solving the convex optimization problem

$$\begin{aligned} \textit{minimize} \quad & \sum_{\tau_i \in \tau_H} (\theta_i^L - u_i^L) \\ \textit{subject to} \quad & \sum_{\tau_i \in \tau_H} \theta_i^H \leq m \\ & \forall \tau_i \in \tau_H, \quad \theta_i^H \geq u_i^H \\ & \forall \tau_i \in \tau_H, \quad \theta_i^H \leq 1 \end{aligned}$$

- $\theta_i^L = \frac{u_i^L . \theta_i^H}{\theta_i^H - u_i^H + u_i^L}$

MCF Scheduling

MCF: Simplified variant of MC-Fluid algorithm

- Rate Assignment:
    - For all HI tasks θ_i^H is given by $\frac{u_i^H}{\rho}$
    - ρ = max {normalized utilization, max $\{u_i^H\}$}
- θ_i^L is computed same way as MC-Fluid
- Linear run-time complexity
    - Compensates on schedulability

Outline

Challenges in Dual-rate MC Fluid Model

Non-optimality: Dual-rate fluid scheduling of MC task systems on multi-core is not optimal

- Feasible task sets are deemed to be not schedulable
  - Example: Multi-rate model
- We cannot extend MC-Fluid or MCF to multi-rate model
  - Complexity of MC-Fluid is high
  - MCF compromises on the schedulability
- Solution: Algorithm with better schedulability and reduced complexity

Outline

MC-Sort algorithm

MC-Sort:

- Maximum rate to a task with a larger HI utilization
- MC-Sort HI rate assignment
  - Assign initial rate of $\frac{u_i^H}{\rho_i}$
    - $\rho_i = max\left\{(\frac{U_H^H}{m}), u_i^H\right\}$
  - Sorts all HI tasks in decreasing HI utilization
  - Assigns maximum rate to tasks in the sorted order until slack remains
- Linearithmic complexity (*i.e.*, *nlogn*)

MC-Sort algorithm

Example: m = 2

Task	T_i	u_i^L	u_i^H	MC-Sort	
				θ_i^L	θ_i^H
τ_1	5	0.3	0.9	-	-
τ_2	7	0.4	0.5	-	-
τ_3	35	0.1	0.3	-	-
τ_4	35	0.45	-	-	-
$\sum$		1.25	1.7	-	-

MC-Sort algorithm

Example: m = 2

Task	T_i	u_i^L	u_i^H	MC-Sort	
				θ_i^L	θ_i^H
τ_1	5	0.3	0.9	-	-
τ_2	7	0.4	0.5	-	-
τ_3	35	0.1	0.3	-	-
τ_4	35	0.45	-	-	-
$\sum$		1.25	1.7	-	-

- Sort all tasks with u_i^H

MC-Sort algorithm

Example: m = 2

Task	T_i	u_i^L	u_i^H	MC-Sort	
				θ_i^L	θ_i^H
τ_1	5	0.3	0.9	-	-
τ_2	7	0.4	0.5	-	-
τ_3	35	0.1	0.3	-	-
τ_4	35	0.45	-	-	-
$\sum$		1.25	1.7	-	-

- Compute $\rho_i = max\left\{(\frac{U_H^H}{m}), u_i^H\right\}$
- $\rho_1 = 0.9 \quad \rho_2 = 0.75 \quad \rho_3 = 0.75$

MC-Sort algorithm

Example: m = 2

Task	T_i	u_i^L	u_i^H	MC-Sort	
				θ_i^L	θ_i^H
τ_1	5	0.3	0.9	-	0.89
τ_2	7	0.4	0.5	-	0.67
τ_3	35	0.1	0.3	-	0.4
τ_4	35	0.45	-	-	-
$\sum$		1.25	1.7	-	1.96

- Initial assignment ($\frac{u_i^H}{\rho}$) is done
- Allocate remaining slack to task with maximum u_i^H

NANYANG TECHNOLOGICAL UNIVERSITY

MC-Sort algorithm

Solution:

Task	T_i	u_i^L	u_i^H	MC-Sort θ_i^L	MC-Sort θ_i^H
τ_1	5	0.3	0.9	0.84	0.93
τ_2	7	0.4	0.5	0.47	0. 67
τ_3	35	0.1	0.3	0.2	0.4
τ_4	35	0.45	-	0.45	-
$\sum$		1.25	1.7	1.96	2.0

- θ_i^L is computed same way as MC-Fluid

NANYANG TECHNOLOGICAL UNIVERSITY

MC-Slope algorithm

- MC-Sort limitation: Does not consider the difference in utilization between criticality levels
  - Task that does maximum execution after mode switch may not get maximum rate allocation

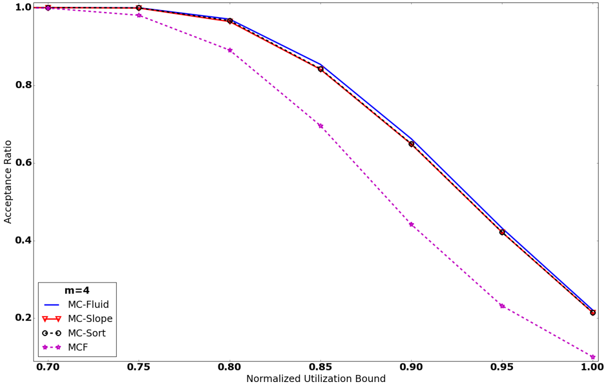
MC-Slope algorithm

MC-Slope: HI rate assignment

- Objective: Minimize $\sum(\theta_i^L - u_i^L)$
- Initial rate: $\theta_i^H = u_i^H$
- Sorts all HI tasks with $R(\theta_i^H)$
  - $R(\theta_i^H) = \frac{d^2(\theta_i^L - u_i^L)}{d\theta_i^{H2}}$
- Assign maximum rate to task with larger $R(\theta_i^H)$
- Linearithmic complexity (*i.e.*, *nlogn*)

Outline

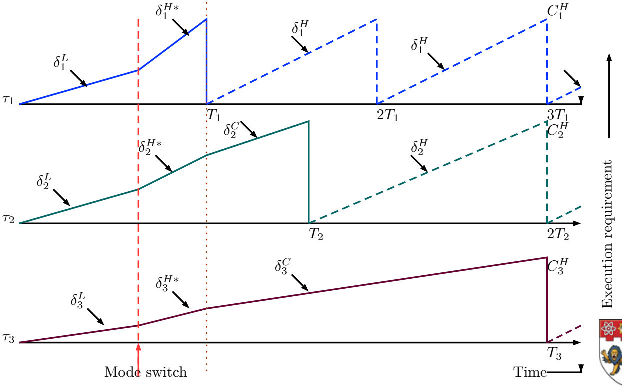
Schedulability



Outline

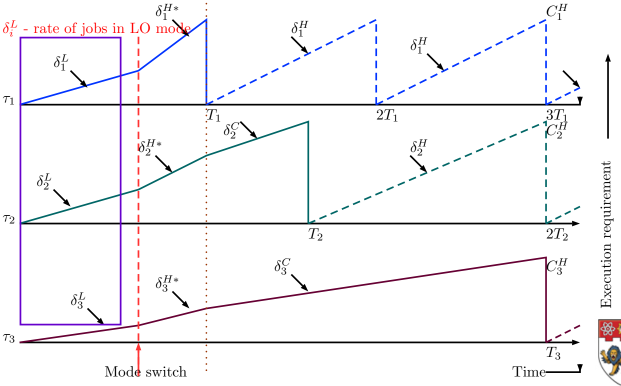
Future Work

Multi-rate model: Each task executes with more than 2 rates



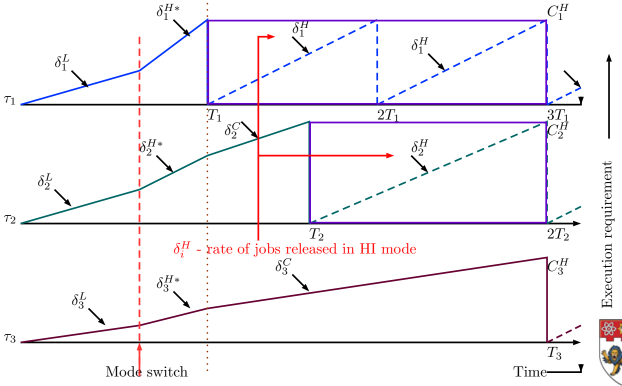
Future Work

Multi-rate model: Each task executes with more than 2 rates



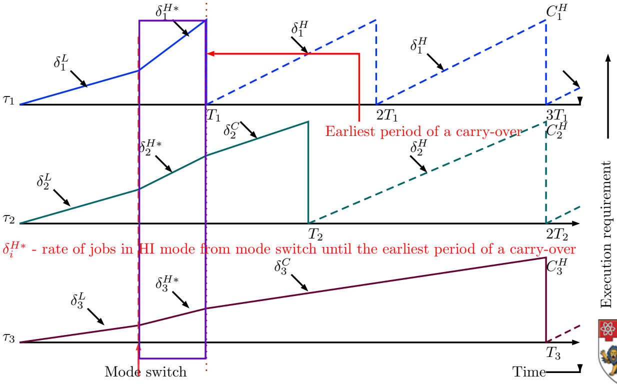
Future Work

Multi-rate model: Each task executes with more than 2 rates



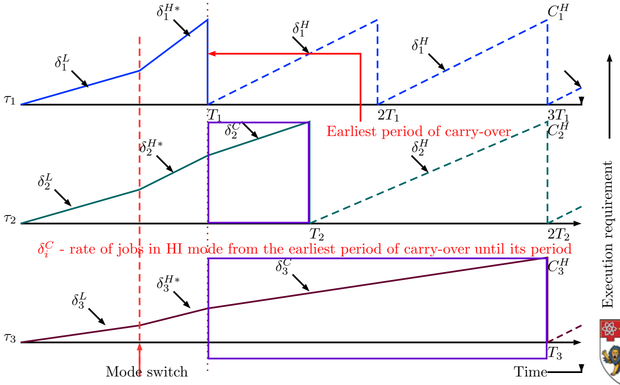
Future Work

Multi-rate model: Each task executes with more than 2 rates



Future Work

Multi-rate model: Each task executes with more than 2 rates



Multi-rate model

Example: m = 2

Task	T_i	u_i^L	u_i^H	MC-Fluid		Multi-rate model			
				θ_i^L	θ_i^H	δ_i^L	δ_i^{H*}	δ_i^C	δ_i^H
τ_1	5	0.3	0.8	0.64	0.94	0.64	0.94	-	0.8
τ_2	7	0.4	0.7	0.70	0.70	0.70	0.70	0.70	0.7
τ_3	35	0.1	0.3	0.22	0.36	0.21	0.36	0.50	0.3
τ_4	35	0.45	-	0.45	-	0.45	-	-	-
$\sum$				2.01	2.0	2.0	2.0	1.2	1.8

Thank you..!

Questions..?

NANYANG TECHNOLOGICAL UNIVERSITY